Using the program discussed here it is possible to implement a general-purpose delay routine with any length between 26 and 281 processor cycles, at a resolution of one cycle.

By W. Couzijn

adjustable delay for PICs

a small but useful routine

If you want to implement wait cycles as part of a microcontroller programming job involving, for example, a Microchip PIC device, most of us would come up with a a simple loop which is executed a predetermined number of times. Using this method you obtain delays whose length is always a multiple of three cycles, a resolution of one cycle being out of the question. By contrast, the routine presented here does offer an accuracy of one cycle within the range 26 to 281 processor cycles.

Source code

Call from the main program:

```
....
MOVLW 50 ;Delay =50
cycles
CALL WAIT ;This line
takes 50 cycles
....
```

```
Assembly code version 1:
WAIT MOVWF BUFFER
  MOVLW 5
  SUBWF BUFFER,F
WAIT1 SUBWF BUFFER,F
  BTFSS BUFFER,0
  BTFSC BUFFER,1
  GOTO WAIT1
WAIT2 SUBWF BUFFER,F
  INCFSZ BUFFER,F
  GOTO WAIT2
  RETURN
```

```
Assembly-code version 2:
WAIT CLRF BUFFER
  SUBWF BUFFER,F
  MOVLW 12
  ADDWF BUFFER,W
  MOVWF BUFFER
  ANDLW 3
  ADDWF PCL,F
  INCF BUFFER,F
  INCF BUFFER,F
  INCF BUFFER,F
  INCF BUFFER,F
  MOVLW 4
WAIT1 ADDWF BUFFER,F
  BTFSS STATUS,Z
  GOTO WAIT1
  RETURN
```

Trick

This routine will be particularly useful for simulating a USART (serial port) on smaller PIC controllers like the PIC12C508/509/671/672 and the PIC16C505. These devices do not have a built-in USART, although an internal R-C oscillator is available. Mind you, the R and C are external components! This oscillator comes in handy because it frees up two extra I/O pins while also avoiding the use of an (expensive) external quartz crystal. However, a problem arises if you want to generate a baudrate of, for example, 9600 bits/s because the on-board R-C oscillator is subject to a tolerance of about 10%.

Fortunately, there's a simple trick to go round this problem. Let's assume a PC connected to the circuit sends data to the PIC in the application circuit. The bit times (i.e., periods) of the received data are then measured. Next, the results of the measurement are used to 'adjust' one of the delay routines in such a way that the same bit times are also employed for transmitted data. In this way, the PIC calibrates itself.

Structure

Two versions have been developed of the general-purpose delay routine. They work as follows.

The desired wait time (delay) is loaded into the working (W) register. Next, the routine called using a CALL instruction. The set delay includes the CALL and RETURN!

Version 1 of the routine is suitable for a delay of between 26 and 281 cycles, with all delays greater than 255 cycles are achieved by setting the desired delay as an 8-bit number. So, '0' for delay of 256 cycles, or '1' for one of 257 cycles, etc.). This routine consists of two loops. In the first loop, the program waits for a multiple of 5 cycles until the delay is divisible by four. Next, the second loop waits a multiple of four cycles. The routine eventually copies a 0 to the working register.

Version2 offers an even shorter minimum delay, the available range being from 17 to 272 cycles. Here, again, 256 = 0, 257 = 1, etc.). The disadvantage of this version is the use of a calculated jump (ADDWF PCF,L) which is not executed properly by every PIC processor in every part of the memory. This should be taken into account!

(990086-1)